



867-10

08/766807
PATENT

SYSTEM AND METHOD FOR GLOBALLY ACCESSING COMPUTER
SERVICES

BACKGROUND OF THE INVENTION

5 1. Field of the Invention

This invention relates generally to computer networks, and more particularly to a system and method for globally accessing computer services.

10 2. Description of the Background Art

The internet currently interconnects about 100,000 computer networks and several million computers. Each of these computers stores numerous application programs for providing numerous services, such as generating, sending and receiving e-mail, accessing
15 World Wide Web sites, generating and receiving facsimile documents, storing and retrieving data, etc.

A roaming user, i.e., a user who travels and accesses a workstation remotely, is faced with several problems. Program designers have developed communication techniques for enabling
20 the roaming user to establish a communications link and to download needed information and needed service application programs from the remote workstation to a local computer. Using these techniques,

the roaming user can manipulate the data on the remote workstation and, when finished, can upload the manipulated data back from the remote workstation to the local computer. However, slow computers and slow communication channels make downloading large files and programs a time-consuming process. Further, downloading files and programs across insecure channels severely threatens the integrity and confidentiality of the downloaded data.

The roaming user alternatively can carry a copy of all potentially needed information and all potentially needed application programs. However, maintaining a true copy of a database is a cumbersome process which often results in a data inconsistency problem. Therefore, a system and method are needed to enable multiple users to access computer services remotely without consuming excessive time, without severely threatening the integrity and confidentiality of the data, and without compromising data consistency.

SUMMARY OF THE INVENTION

08766307:121306

The present invention provides a system and method for communicating through a computer network. The system includes a master server, service engines and clients, each equipped with a communication engine for establishing communications links. The system further includes a browser for receiving applet information corresponding to a service performed by one of the service engines from the server, and an applet engine for using the applet information to control user interface I/O of the service. The master server may confirm access privileges before allowing clients to access a service. The communications engine receives configuration data which are used to configure the client components such as the operating system, the web browser, the applets, etc. to enable communication with the service engine and to provide a user-specific user interface to the client. It will be appreciated that the system may be stored on a floppy disk or hard drive.

The method includes establishing a communications link between a client and a server, receiving applet information by the client from the server, establishing a communications link between the client and a service using the applet information, and directing user interface I/O control of the service to the client using the applet information. Establishing a communications link may include

confirming access privileges. The method may further include receiving configuration data, and using the configuration data for configuring the client components such as the operating system, the web browser, the applets, etc. to enable communication with the
5 service engine and to provide a user-specific user interface to the client.

The system and method use web browser technology to enable multiple users to access computer services remotely. Furthermore, the system and method use applet technology to enable users to
10 access computer services without consuming excessive time, without severely threatening the integrity and confidentiality of the data, and without compromising data consistency. Still further, the system and method use configuration data to enable users to obtain a consistent interface from any remote terminal.

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. 1 is a block diagram illustrating a roaming-user network access system in accordance with the present invention;

FIG. 2 is a block diagram illustrating details of the remote client
5 of FIG. 1;

FIG. 3A is a block diagram illustrating details of the master server of FIG. 1;

FIG. 3B is a block diagram illustrating details of the configuration data of FIG. 3A;

FIG. 4 is a block diagram illustrating details of each local client
10 of FIG. 1;

FIG. 5 illustrates an example graphical user interface of web page as maintained by the web page engine of FIG. 1;

FIG. 6 is a flowchart illustrating a method for globally accessing
15 a particular computer service;

FIG. 7 is a flowchart illustrating details of the FIG. 6 step of initiating execution of a service on the local client;

FIG. 8 is a flowchart illustrating alternative details of the FIG. 6 step of initiating execution of a service on the local client; and

FIG. 9 is a flowchart illustrating details of the FIG. 6 step of
20 providing service access to the remote client user.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENT

FIG. 1 is a block diagram illustrating a roaming-user network access system 100, in accordance with the present invention. System 100 includes a network of computers, referred to herein as an "internet" 140, connected via a master server 130 to a local network 120. A first local client 110 and a second local client 160 are addressably connected via the local network 120 to the master server 130. A remote client 150 is connected via the internet 140 to the master server 130. Another server 166 is also connected via the internet 140 to the remote client 150, to the master server 130 and to the local clients 110 and 160.

The first local client 110 includes a first service engine 115 for providing a first computer service, such as calendaring, e-mail, paging, word-processing or the like, to a user. The second local client 160 includes a second service engine 165 for providing a second computer service to a user. Service engines 115 and 165 are described in greater detail with reference to FIG. 4. The remote client 150 includes a client engine, an example of which is an applet-based web browser 155 which is configured for communicating with the internet 140 and for reading and executing applets. Examples of web browsers 155 include the Netscape™ web browser produced by the Netscape Corporation, the Internet Explorer™ web browser

produced by the Microsoft Corporation, or the Java™ Developers Kit 1.0 web browser produced by Sun Microsystems, Inc. While the illustrated client engine is a web browser 155, those skilled in the art will recognize that other client engines could be used in accordance with the present invention. ^{Although the embodiments herein are} ~~Although "applets" are~~ being described with reference to the Java™ distributed environment produced by Sun Microsystems, Inc., ^{"Downloadables" herein} ~~"applets"~~ correspond to all downloadable and executable or interpretable programs for use in a distributed environment such as in the ^{ActiveX™ controls} ActiveX™ distributed environment produced by the Microsoft Corporation.

The master server 130 includes a web page engine 133 for maintaining and providing access to an internet web page which is enabled to forward applets 136 to the web browser 155 of the remote client 150. A user can select a particular service which corresponds to one of the applets 136, thereby causing the web browser 155 of the remote client 150 to initiate execution of the corresponding applet 130. Execution of the applet 130 causes the appropriate service engine 115 or 165 to execute the particular service or access an already executing service. It will be appreciated that executing a service may include directing Input/Output (I/O) control of a service application program, such as an e-mail

application program, a paging application program or a word-processing application program, to the remote client 150 user.

The master server 130 further includes configuration data 137, which the remote client 150 can download to gain access to the
5 desired service and to configure the functionality, look and feel of the web browser 155. The configuration data 137 may include operating system settings such as TCP protocol data and the domain name server address, user preferences, bookmarks, services, service addresses, etc. Each user preferably uploads unique configuration
10 data 137 to the master server 130 in order to obtain similar functionality, look and feel from any web browser 155.

Similar to the local clients 110 and 160, the master server 130 may include a service engine 138 for providing access to a third computer service, and server 166 may include a service engine 167
15 for providing access to a fourth computer service. Service engines 138 and 166 are described in greater detail with reference to FIG. 4. From the remote client 150 user's standpoint, the location of the service engine does not matter.

As shown by line 1, the remote client 150 user inputs a
20 predetermined Uniform Resource Locator (URL) address for opening the web page managed by the web page engine 133 of the master server 130. As shown by line 2, the web page engine 133 sends at

least one of the applets 136 and web browser configuration data 137 to the web browser 155. From the services offered by the web page, the user selects a particular service corresponding to one of the applets 136. For example, to access e-mail services, the user selects
5 an e-mail prompt corresponding to an e-mail applet 136.

As shown by line 3, the web browser 155 forwards to the master server 130 a service request for initiating execution of the selected service. As shown by line 4, the master server 130 determines the location of the service and sends the service request
10 to the appropriate address, i.e., either to the service engine 115 of the local client 110, to the service engine 165 of the local client 160, to the service engine 138 of the master server 130 or to service engine 167 of the other server 166.

To initiate execution of the service request, the master server
15 130 may use servlets or agents. The appropriate service engine 115, 138, 165 or 167 executes the service and generates responsive data accordingly. As shown by line 5, the service engine 115, 138, 165 or 167 sends the responsive data back to the master server 130 and, as shown by line 6, the master server 130 sends the responsive data
20 back to the remote client 150.

Since the master server 130 stores the applets 136 and the configuration data 137, the roaming user can access a service via any

computer 110, 130, 150 or 160 which includes an applet-based web browser 155 and which is connected to the master server 130.

Further, because the master server 130 maintains the network addresses of all services, the user need only maintain the URL

5 address of the web page maintained by the web page engine 133.

That is, the user need not maintain the current address of each service.

Although not shown in FIG. 1, a web browser 155 may be placed on a local client 110 or 160. A user can access the web page engine 133 from the local client 110 or 160 to take advantage of the services provided by the master server 130, by the server 166 or even by the local clients 110 or 160. For example, the local client user may access the web page engine 133 and select a particular service which resides on the local client 110. Accordingly, the applets 136 would launch the service on the local client 110 and would provide direct I/O control of the service to the user. Those skilled in the art will recognize that a particular service engine may require the services of other service engines. It will be appreciated that the local client user need not know the location of each service.

20 The local client user need only access the master server 130.

FIG. 2 is a block diagram illustrating details of the remote client 150, which includes a Central Processing Unit (CPU) 210 such as a Motorola Power PC® microprocessor or an Intel Pentium® microprocessor. An input device 220 such as a keyboard and mouse, and an output device 230 such as a Cathode Ray Tube (CRT) display are coupled via a signal bus 240 to CPU 210. A communications interface 250, a data storage device 260, such as Read Only Memory (ROM) and a magnetic disk, and a Random-Access Memory (RAM) 270 are further coupled via signal bus 240 to CPU 210. The communications interface 250 is coupled to the internet 140.

An operating system 280 includes a program for controlling processing by CPU 210, and is typically stored in data storage device 260 and loaded into RAM 270 for execution. Operating system 280 further includes a communications engine 282 for generating and transferring message packets to and from the internet 140 via the communications interface 250. Operating system 280 further includes an Operating System (OS) configuration engine 284, which configures the operating system 280 based on OS configuration data 137 such as TCP protocol data, domain name server addresses, etc. received from the master server 130.

Operating system 280 further includes a web browser 155 as described with reference to FIG. 1. The web browser 155 includes a

Web Browser (WB) configuration engine 288, which configures the web browser 155 based on WB configuration data 137 such as home page addresses, bookmarks, caching data, user preferences, etc.

received from the master server 130. The web browser further

5 includes an applet engine 290 for handling the execution of downloaded applets 294. The applet engine 290 includes an Applet Engine (AE) configuration engine 292, which configures the applet engine 290 based on AE configuration data 137 received from the master server 130. The applet engine 290 further includes applet-specific configuration data 296 received from the master server 130. For example, for the e-mail applet 294, e-mail applet configuration data 296 may include a user's e-mail address, name, preferred signature block, as well as customizable user interface parameters.

15 FIG. 3A is a block diagram illustrating details of the master server 130, which includes a Central Processing Unit (CPU) 310 such as a Motorola Power PC[®] microprocessor or an Intel Pentium[®] microprocessor. An input device 320 such as a keyboard and mouse, and an output device 330 such as a Cathode Ray Tube (CRT) display
20 are coupled via a signal bus 340 to CPU 310. A communications interface 350, a data storage device 360, such as Read Only Memory (ROM) and a magnetic disk, and a Random-Access Memory (RAM)

370 are further coupled via signal bus 340 to CPU 310. The communications interface 350 is coupled between the internet 140 and the local network 120.

An operating system 380 includes a program for controlling
5 processing by CPU 310, and is typically stored in data storage device 360 and loaded into RAM 370 for execution. Operating system 380 further includes a communications engine 382 for generating and transferring message packets via the communications interface 350 to and from the internet 140 and to and from local network 120.
10 Operating system 380 also includes a web page engine 133 for maintaining a web page and for forwarding the applets 136 and the configuration data 137 to the remote client 150.

Operating system 380 further includes an applet host engine 383 for managing and responding to the execution of downloaded
15 applets 288. It will be appreciated that, to respond to the execution of the applets 288, the applet host engine 383 may execute corresponding servlets 384 or forward corresponding agents 386 for execution on the appropriate local client 110 or 160. Operating system 380 may further include the service engine 138 for providing
20 the third service to the user. Service engine 138 is described in greater detail with reference to FIG. 4.

14

FIG. 3B is a block diagram illustrating details of the configuration data 137. Configuration data 137 includes operating system 280 settings 392 such as TCP protocol data and the domain name server address; web browser 155 settings 392 such as home page address, bookmarks and caching data; applet engine 290 settings 392; and applet configuration data 296 such as the user's e-mail address, name and signature block. It will be appreciated that applet-specific configuration data 296 is needed since the service may not be located on the user's own local client 110 or 160. Configuration data 137 further includes predetermined user preferences 394 such as font, window size, text size, etc.

Configuration data 137 further includes the services 396 which will be provided to the user. Services 396 include a list of registered user's and each user's desired services 396, which specify each user's set of downloaded applets 294. Configuration data 137 further includes service addresses 398 specifying the location of each of the services 396 accessible via the master server 130.

FIG. 4 is a block diagram illustrating details of a local client 400, which includes a Central Processing Unit (CPU) 410 such as a Motorola Power PC® microprocessor or an Intel Pentium®

microprocessor. Local client 400 is illustrative of local clients 110 and 160. An input device 420 such as a keyboard and mouse, and an output device 430 such as a Cathode Ray Tube (CRT) display are coupled via a signal bus 440 to CPU 410. A communications interface
5 450, a data storage device 460, such as Read Only Memory (ROM) and a magnetic disk, and a Random-Access Memory (RAM) 470 are further coupled via signal bus 440 to CPU 410. The communications interface 450 is coupled to the local network 120.

An operating system 480 includes a program for controlling
10 processing by CPU 410, and is typically stored in data storage device 460 and loaded into RAM 470 for execution. Operating system 480 also includes a communications engine 481 for generating and transferring message packets via the communications interface 450 to and from the internet 140 and to and from local network 120.
15 Operating system 480 further includes a service engine 490, which includes a service interface 482 for receiving and translating service requests generated by the applet 136 corresponding to the user-selected service, a service processor 484 and service data 484 for executing the service request. The service data 484 may include
20 previously-generated documents, database information, etc. It will be appreciated that service engine 490 is similar to the service engine 115, 138, 165 or 167 of local client 110, of master server 130,

of local client 160 or of other server 166. Those skilled in the art will recognize that operating system 480 may optionally include a web browser similar to web browser 155 as described with reference to FIG. 2, even though no web browser is shown within
5 operating system 480.

FIG. 5 illustrates an example graphical user interface 500 of a URL-addressable HyperText Markup Language (HTML)-based web page, as maintained by the web page engine 133 of the master
10 server 130. The graphical user interface 500 includes a title 510 "Roam Page," a listing of the provided services 396 and a pointer 570 for selecting one of the provided services 396. As illustrated, the provided services may include an e-mail service 520, a calendaring service 530, an internet access service 540, a paging service 550 and
15 a fax sending service 560. Although not shown, other services such as bookmaking, QuickCard™, etc. may be included on the web page.

FIG. 6 is a flowchart illustrating a method 600 for globally accessing a particular computer service. Method 600 begins with
20 step 610 by the web browser 155 using a URL to open a communications link of the remote client 150 to the web page engine 133 of the master server 130. Upon opening the communications

link, the web browser 155 in step 620 downloads configuration data 388 from the master server 130, which in step 630 the remote client 150 uses to configure itself. Step 630 includes using the OS configuration engine 282 to configure the operating system 280, 5 using the WB configuration engine to configure the web browser 155, and using the AE configuration engine 288 to configure the applet engine 290.

0876307-121496
96727-1099230

The applet engine 290 of web browser 155 in step 640 downloads at least one applet 294 from the applets 136 of master 10 server 130. Based on the configuration data 137 and on the downloaded applets 288, the web browser 155 in step 650 generates and displays a web page on the remote client output device 230. The remote client 150 user in step 660 can use the remote client input device 220 to select a particular one of the services 396 offered by 15 the web page. Selection of a service causes the applet engine 290 to connect to or execute the applet 294 corresponding to the selected service 396.

The corresponding applet 294 in step 670 causes the appropriate service engine 490 to execute the selected service 396. 20 Step 670 is described in greater detail with reference to FIGs. 7 and 8. The appropriate service engine 490 in step 680 executes the service request and accordingly provides service access to the

remote client 150 user. Step 680 is described in greater detail with reference to FIG. 9.

FIG. 7 is a flowchart illustrating details of step 670, which

- 5 begins with step 710 by the corresponding applet 294 forwarding a service request to a corresponding servlet 384 on master server 130. The servlet 384 in step 720 compares the service request with the service addresses 398 stored in the master server 130 to retrieve the service address 398 corresponding to the requested service 396.
- 10 The servlet 384 in step 730 uses the service address 398 to forward the service request to the service interface 482 of the appropriate service engine 490. The service interface 482 in step 740 receives and initiates execution of the service 396 by the appropriate service engine 490.
- 15 The service interface 482 in step 750 informs the servlet 384 of the initiation of the service 396. Accordingly, the servlet 384 in step 760 acts as a proxy for the executing applet 294 and in step 770 informs the applet 294 of the initiation of the service 396. The applet 294 in step 780 acts as the Input/Output (I/O) interface to the
- 20 service 396 (via the servlet 384). Acting as I/O interface includes using the remote client 150 output device 230 to display responsive data received from the service engine 490, using remote client 150

input device 220 to forward instructions to the service engine 490, and using the communications engine 282 and the communications interface 250 to packet and exchange data with the service engine 490.

5

FIG. 8 is a flowchart illustrating alternative details of step 670, which begins with step 810 by applet 294 requesting the master server 130 to open a direct communications link with the appropriate service engine 490. The master server 130 in step 820
10 compares the service request with service addresses 398 stored in the master server 130 to retrieve the proper service address 398 and sends the service address and other service-specific configuration data to the applet 294. The remote client 150 in step 830 opens the direct link to the service.

15 When the direct connection has been established, the applet 294 in step 840 forwards the service request to the service interface 482 of the service engine 490. The service interface 482 in step 850 causes the service engine 490 to initiate execution of the requested service 396, and in step 860 informs the applet 294 of the initiation
20 of the service 396 by the service engine 490. The applet 294 in step 870 acts as the I/O interface to the service 396.

20

FIG. 9 is a flowchart illustrating details of step 680, which begins with step 910 by the CPU 310 or 410 of the master server 130, the local client 110 or the local client 160 loading the respective service engine 490 into RAM 370 or 470. The CPU 310 or 410 in step 920 initiates execution of the service engine 490, which in step 930 processes the service request and accordingly generates and forwards display data and I/O requests as responsive data to the remote client 150. For example, if the user requests access to a database of information, the service engine 490 in step 930 will retrieve and send the requested data and a prompt for additional instructions back to the user. The service engine 490 in step 940 waits for instruction from the remote client 150 user. As illustrated by step 960, steps 920-960 repeat until the user instructs the system to quit.

For example, if a user wants to send an e-mail message to a colleague but maintains an e-mail application program and an e-mail address database only on his local work computer, then using the system of the present invention the user can operate the e-mail application program and can access the e-mail address database from a remote computer 150. That is, either a master server 130 or the

local work computer 110 includes a web page engine 133 maintaining a roaming-access web page. The user locates a remote computer 150 operating an applet-enabled web browser 155, and inputs the roaming-access web page URL address.

5 The web page engine 133 confirms the user's access privileges. If access is granted, then the web browser 155 downloads information for displaying the web page, information for configuring the operating system 280 and the web browser 155 according to the user's predetermined specifications. The web browser 155 further
10 downloads a predetermined set of service applets 294 which include an e-mail applet 294. The user selects the e-mail applet 294, for example, by clicking a mouse button while the mouse pointer 570 is over an e-mail applet prompt 520. Accordingly, the web browser 155 executes the e-mail applet 294, which includes retrieving the e-
15 mail application program address 398 and sending an e-mail access service request to the service interface 482. The service interface 482 initiates the e-mail application program 484 and directs Input/Output (I/O) control to the remote computer 150 user.

20 The foregoing description of the preferred embodiments of the invention is by way of example only, and other variations of the above-described embodiments and methods are provided by the

present invention. For example, although the remote client 150 of the has been described as a personal computer, the remote client 150 can alternatively be a Java™ workstation or a network computer.

Although the remote client 150 has been described as downloading

5 all selectable applets 294 from the master server 130, the remote client 150 can alternatively download selectable service prompts and can then download from the master server 130 only the applet

corresponding to the selected service. Further, although the remote

client 150 has been described as downloading all the configuration

10 data 137 in one request, the remote client 150 may use several

requests to download the configuration data 137. Still further,

although services are described as being performed by one service

engine 490, a service may be provided by multiple service engines

490. Still even further, components of this invention may be

15 implemented using a programmed general purpose digital computer,

using application specific integrated circuits, or using a network of

interconnected conventional components and circuits. The

embodiments described herein have been presented for purposes of

illustration and are not intended to be exhaustive or limiting. The

20 system is limited only by the following claims: